

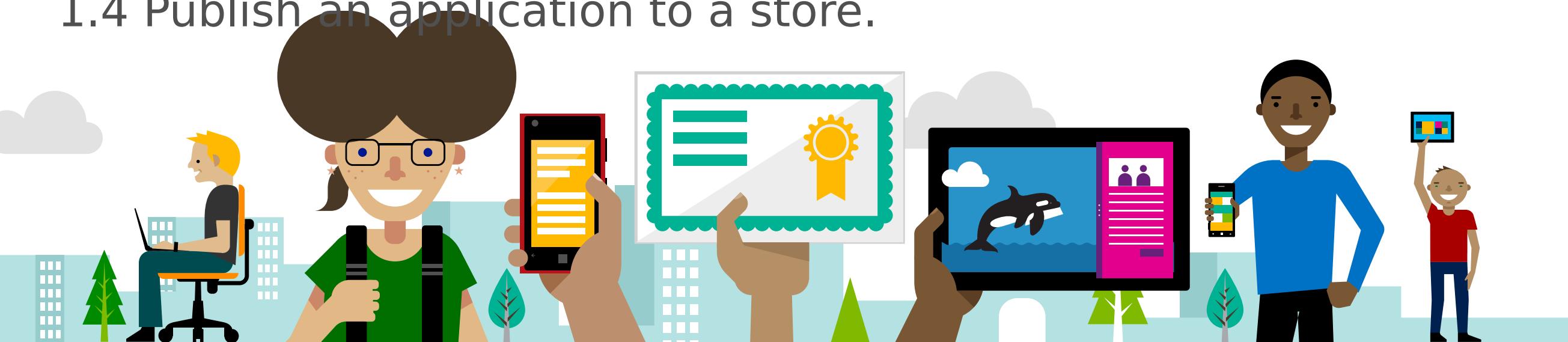
The Application Lifecycle

1.1 Understand the platform fundamentals.

1.2 Manage the state of an application.

1.3 Debug and test an HTML5-based touch-enabled application.

1.4 Publish an application to a store.



Agenda

1 Steps of App Building

2 The Runtime Environment

3 Managing App Data

4 Debug and Test an App

5 Validate an App

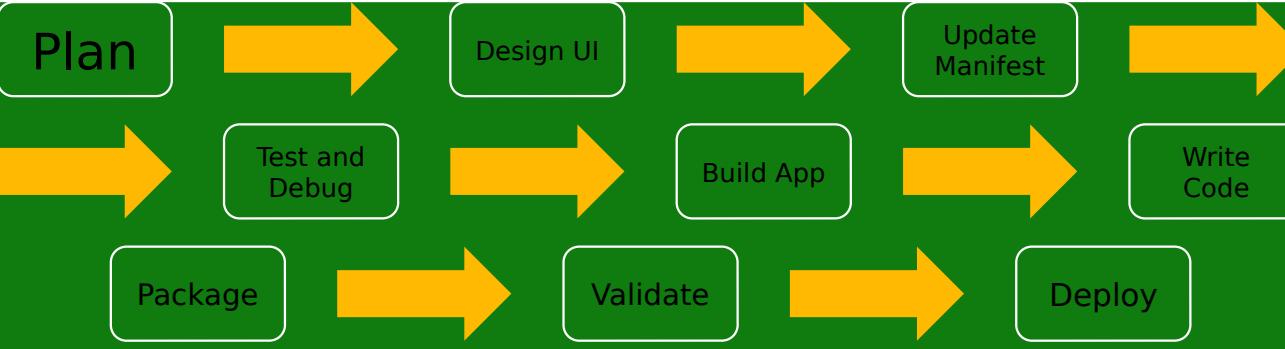
6 Publish an App to a Store

7 Packaging Apps



Steps of App Building





How to Build an App

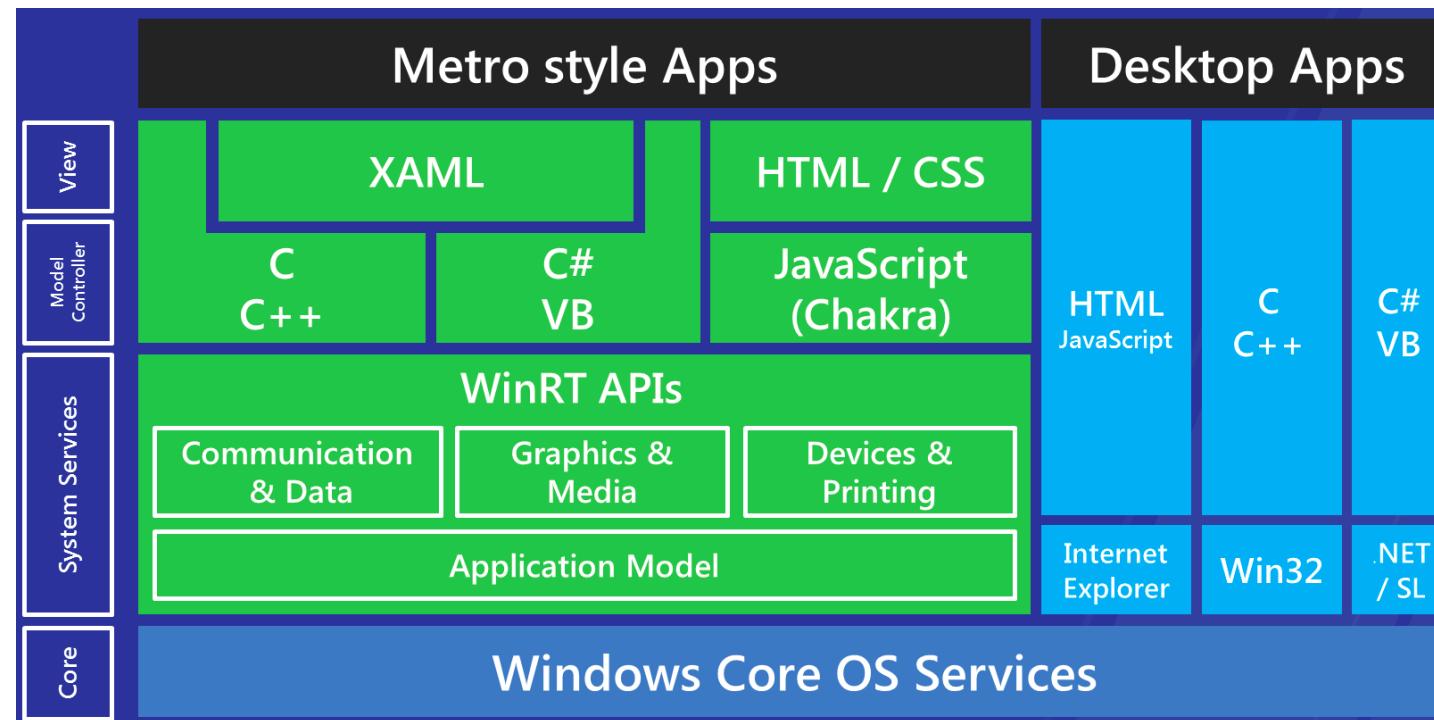
- Creating an app requires a number of steps, including:
 1. Planning your project
 2. Designing the user interface (UI)
 3. Updating the app manifest
 4. Writing the code
 5. Building the app
 6. Debugging and testing the app
 7. Packaging the app
 8. Validating the app
 9. Deploying the app

The Runtime Environment



Runtime Environments

- When you launch an app, it's considered to be in a **runtime environment (RTE)**
- An RTE is where users run the app and developers test it
- In Windows, this environment is called **Windows Runtime (WinRT)**



Windows Runtime (WinRT)

- Provides developers with access to a user's device
 - Hardware
 - Operating System
- It does this through WinRT and Windows Library for JavaScript APIs
 - An **application programming interface (API)** is a set of guidelines that allow one program to communicate with another

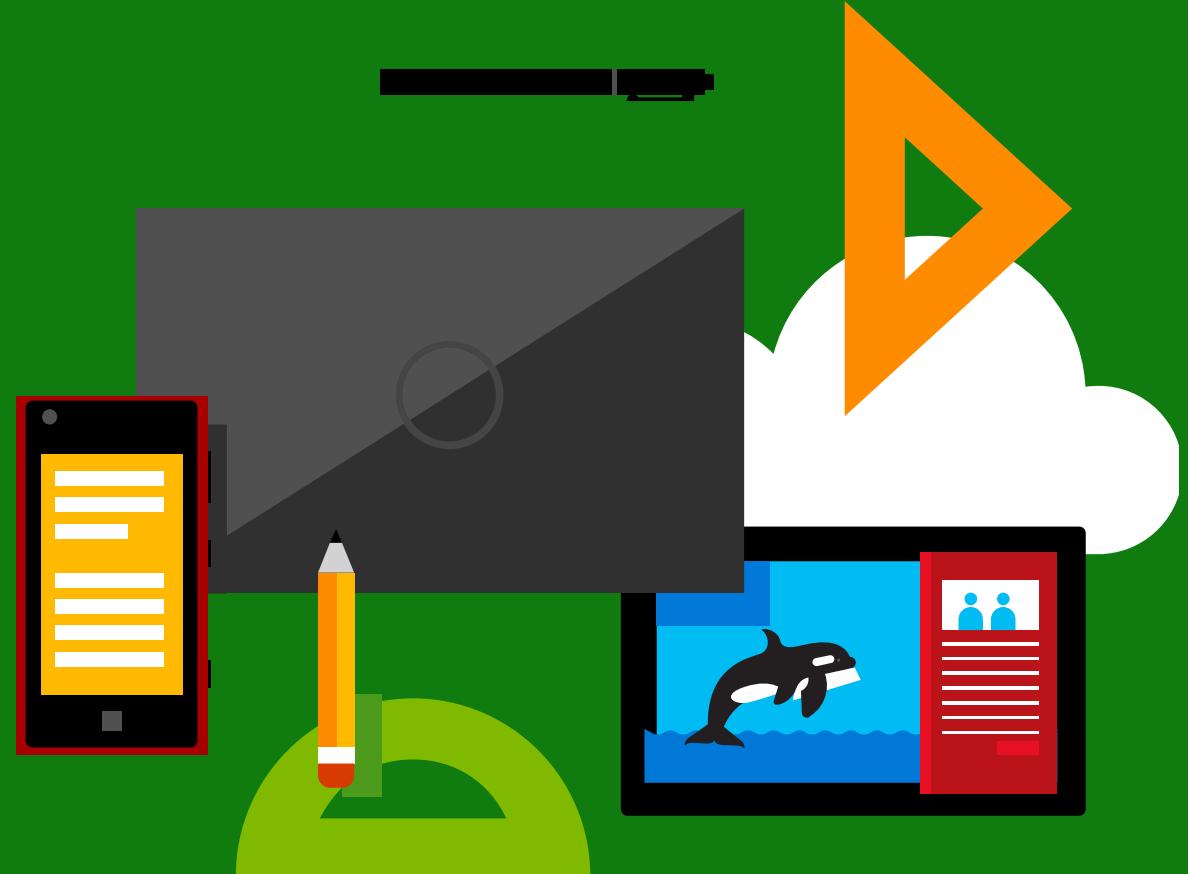
The App Container

- When a Web app executes, it does so in a contained environment
 - An app container is a separate memory space
 - The purpose of an app container is to prevent apps from corrupting the operating system

.NET Framework

- Provides a secure environment for Web apps to run
- The framework uses security **transparency** to separate different kinds of code while running
 - Transparency prevents app code from running with infrastructure code
- **Permission sets** define what application code has access to and ability to do

Managing App Data



Accessing a Web Page

- When we enter a **uniform resource locator (URL)** into the address bar of a browser, the browser sends an **HTTP request** to a Web server for a Web page
 - HTTP is short for **Hypertext Transport Protocol**
 - HTTP is a **stateless protocol**, which means that it doesn't retain data from session to session
- Therefore, when we close a Web browser after using an application, data is not automatically saved



State Management

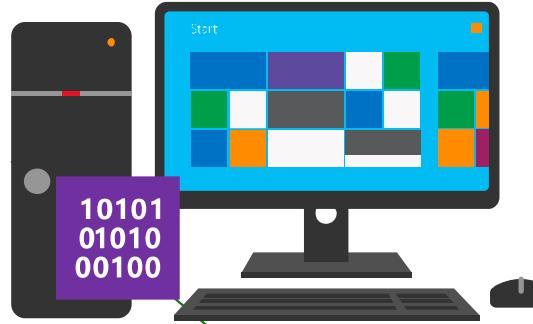
- When users request access to an application, or use a Web browser, a **state** is created
- There are three different types of states, including:
 - **session states**
 - **application states**
 - **persistent states**
- When a state starts and ends is dependent upon which type of state it is
- **State management** is the process of maintaining Web page information about a web page

Different States

- A **session state** is created when access to an application is requested by logging in
 - It ends when a user ends the session, or logs out of the app
- An **application state** is created when a browser sends a request for a Web page to a web server
 - It ends when a user closes the browser
- **Persistent state information** is data that remains for use by an application after a session ends
 - This allows an app to continue its state when a user returns to a site

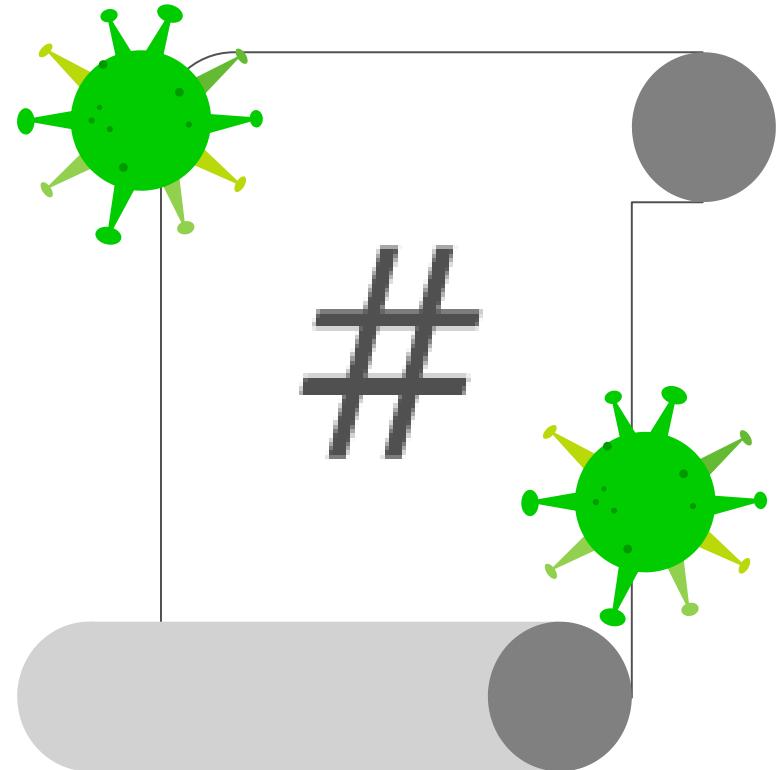
Cookies

- To work around the fact that HTTP doesn't retain data from session to session, developers can use **cookies**
 - Cookies are small files that save information about users that are saved on a user's computer
 - Browsers send cookies back to Web servers
 - Servers use the files to identify a user and customize their experience



Limitations of Cookies

- The use of cookies can present a number of problems:
 - Security risks
 - Performance decreases due to the amount of data sent back and forth between computers
- With HTML5, developers can use Web storage instead
- There are two types of Web storage:
 - `localStorage`
 - `sessionStorage`



localStorage and sessionStorage

- **localStorage** lets users save larger amounts of persistent data
 - There is no limit to how long the data persists
- **sessionStorage** lets users save session state data
 - It only lasts for the duration of the session
- Both methods allow users to store large amounts of data without slowing down a connection because data is transferred only when requested

AppCache for Offline Files

- Data can be stored locally when a user is offline using the **Application Cache**, or **AppCache**
- AppCache stores resources, such as HTML, CSS, and JavaScript files, locally on a user's machine
 - As a result, users can access Web pages and apps offline
- Developers can dictate which type of information is stored locally in the **cache manifest file**

Debug and Test an App



Debugging and Testing

- **Debugging** and **testing** are incredibly important steps in developing high-quality Web applications because they ensure apps run reliably and error-free
- Debugging is the process of detecting, finding, and correcting errors in **logic** and **syntax**
 - **Logic errors** prevent the app from behaving as expected
 - **Syntax errors** are typos in code which prevent Web apps from running

Validate an App



Validating HTML5 Code

- An important aspect of debugging and testing is to validate code to ensure it is properly interpreted by browsers
- We can use a **validator** to test code for inaccuracies and syntax errors
- **W3C** provides a code validation service for all versions of HTML and CSS
 - Access the Markup Validation Service Web page at <http://validator.w3.org/>

Packaging Apps



Packaging Apps

- Packaging an app is the process of preparing an app for installation on different devices or systems
- If you want to package and publish an app in the Windows Store, you can use the **Windows App Certification Kit** to test it
 - The Windows App Certification Kit provides a report describing any problems with an app
 - If there are problems, then you should correct them and test your app again until it works

Publish an App to a Store



Publishing an Application

- After you test, debug, and validate your code, it will be ready for upload to a marketplace for apps
- If the app was built for Windows, then the marketplace will be the Windows Store
- Before publishing an app to the Windows Store, you must:
 1. Sign up and pay for a developer account
 2. Go through the app submission checklist
 3. Capture screenshots of unique features in your app
 4. Have other users test your app on multiple devices and platforms
 5. Include a privacy statement if your app gathers personal information or copyrighted software

Summary

| | | | |
|---|-------------------------|---|---------------------------|
| 1 | Steps of App Building | 6 | Publish an App to a Store |
| 2 | The Runtime Environment | 7 | Packaging Apps |
| 3 | Managing App Data | | |
| 4 | Debug and Test an App | | |
| 5 | Validate an App | | |

